

Задание 2. Параллельный поиск в файлах.

Вы никогда не теряли информацию на своём компьютере? Не было такого, что вы точно знаете, что она где-то есть, но в каком файле — не помните? Вы помните только кусочек из текста, который вы набирали и, возможно, директорию с вашими проектами, в глубине которой хранился этот файл. Поможем сами себе, написав простой поисковик.

Программа должна получить из командной строки то, что вы будете искать (pattern) — обыкновенную текстовую строку, для простоты содержащую только символы из нижней половины таблицы ASCII (латинские буквы, цифры, знаки препинания и прочее). Кириллических и прочих символов мы искать не будем (если хотите, попробуйте, но я этого не требую). Другой аргумент — имя директории, с которой поиск начинается. Программа должна распознавать также число вычислительных потоков, которые будут заниматься операциями поиска.

Программа должна обрабатывать два флага.

- `-t#` — поиск в `#` потоков.
- `-n` — поиск только в указанной директории.

Эти флаги могут находиться в любом месте командной строки. При их отсутствии программа осуществляет поиск в 1 поток и в указанной директории и всех её поддиректориях.

Примеры вызова:

- `./psearch -t4 FILE /usr/include`
Искать образец `FILE` в директории `/usr/include` и глубже в 4 потока.
- `./psearch FILE`
Искать образец `FILE` в текущей директории и глубже в 1 поток.
- `./psearch FILE -t4`
Искать образец `FILE` в текущей директории и глубже в 4 потока.
- `./psearch FILE -n -t4`
Искать образец `FILE` только в текущей директории в 4 потока.

Для каждой найденной строки с образцом нужно вывести имя файла, в котором она была найдена, её номер и её саму. Не выводите одну и ту же строку несколько раз.

Максимальную оценку получает решение, которое быстрее всего выдаст результат на заданном наборе файлов и строке поиска. Желательно использовать хороший алгоритм поиска подстрок в строках. Идеально — построить конечный автомат, но, видимо, я много хочу...

Постарайтесь задействовать все потоки в полную силу. В идеальном решении простаивания потоков не должно быть или оно должно быть как можно меньшим.

Последний срок сдачи задачи №2 — 20 ноября.